

# PRACTICAL RECORD FILE

---

**NAME** :

**ADMISSION NO.** :

**ROLL NUMBER** :

**CLASS** :

**SUBJECT** : COMPUTER SCIENCE(Code – 083)

**YEAR** : 2020-21

**SCHOOL** : DELHI PUBLIC SCHOOL SURAT

S. NO.	PROGRAM	PAGE NO.
1	<p>Write a program to input a start and stop value and do the following on user choice:</p> <ol style="list-style-type: none"> <li>1. Print all prime numbers in the range</li> <li>2. Print all perfect numbers in the range</li> <li>3. Print all special numbers in the range</li> <li>4. Print all the palindromic number in the range</li> </ol>	6
2	Write a program to input a tuple and print the sum of those values in the tuple, which are ending with 0 or 5.	8
3	<p>Write a program to input a tuple of integer elements and do the following on user choice:</p> <ol style="list-style-type: none"> <li>1. Print the reverse of each tuple element</li> <li>2. Print the factorial of first digit of each tuple element</li> <li>3. Print the maximum and minimum integer in the tuple</li> </ol>	9
4	<p>Write a program to input a list of integer elements and do the following on user choice:</p> <ol style="list-style-type: none"> <li>1. Print the sum of all the odd elements in the list</li> <li>2. Remove duplicates from the list</li> <li>3. Print the cube of all list elements which are divisible by 2 and 3</li> </ol>	11
5	<p>Write a menu driven program to input a matrix and do the following on user's choice:</p> <ol style="list-style-type: none"> <li>1. Print sum of each row</li> <li>2. Print sum of each column</li> <li>3. Print all the diagonal elements</li> </ol>	13
6	<p>Write a menu driven program to input a matrix and do the following on user's choice:</p> <ol style="list-style-type: none"> <li>1. Print upper half of matrix</li> <li>2. Print lower half of matrix</li> <li>3. Print middle row and middle column elements</li> </ol>	16
7	<p>Write a menu driven program to input a matrix and do the following on user's choice:</p> <ol style="list-style-type: none"> <li>1. Swap first row elements with last row elements</li> <li>2. Swap first column elements with last column elements</li> </ol>	19
8	<p>Write a program to input a square matrix. Print the sum of alternate elements.</p> <p>For example :</p> <p>Input matrix :      1  2  3                       4  5  6                       7  8  9</p> <p>In the given matrix sum will be calculated for <math>1 + 3 + 5 + 7 + 9</math></p> <p>Output :             Sum of alternate elements = 25</p>	22
9	Write a Python program to input two sorted list n ascending order and merge them to form a new list in ascending order.	24
10	Write a Python program to input two sorted list n descending order and merge them to form a new list in ascending order.	25
11	<p>Write a program to input a tuple and sort it based on user choice:</p> <ol style="list-style-type: none"> <li>1. Using Bubble sort</li> <li>2. Using insertion sort</li> </ol>	26

<b>12</b>	Write a program to input a string and do the following on user's choice: 1. Print all four letter words 2. Print all palindromic words 3. Print all the words starting with 'a' or 'A' 4. Print all the words which starts and ends with a vowel	<b>28</b>
<b>13</b>	Write a program to find the number of vowels, consonants, digits and white space characters in a string.	<b>31</b>
<b>14</b>	Write a Python program to check the validity of password input by users. Validation : <ul style="list-style-type: none"> <li>• At least 1 letter between [a-z] and 1 letter between [A-Z].</li> <li>• At least 1 number between [0-9].</li> <li>• At least 1 character from [\$#@].</li> <li>• Minimum length 6 characters.</li> <li>• Maximum length 16 characters.</li> </ul>	<b>32</b>
<b>15</b>	Write a program to check if two strings are anagram or not. NOTE: An anagram of a string is another string which contains all the characters of original string in different sequence. Example "abcd" and "dcba" are anagram.	<b>34</b>
<b>16</b>	Write a program that repeatedly asks the user to enter product name and price. Store all of these in a dictionary whose keys are Product name and value is the price. From the entered dictionary print average price and product name which has the highest price.	<b>35</b>
<b>17</b>	Create a dictionary whose keys are month names and whose values are the number of days in the corresponding months. 1. Ask the user to enter a month name and use the dictionary to tell how many days are there in the month. 2. Print all the keys in alphabetical order. 3. Print names of all the months with 31 days	<b>36</b>
<b>18</b>	Write a program to create a text file "Diary.txt" and do the following on user choice: 1. Display the number of vowels 2. Display the number of words 3. Display the number of lines 4. Display the most common word in file	<b>38</b>
<b>19</b>	Write a menu driven program that reads a text file "data.txt" and do the following on user's choice: 1. Display total number of uppercase alphabets in the file 2. Display how many times "the" word is present in the file 3. Display total number of lines starting with an uppercase vowel	<b>40</b>
<b>20</b>	Write a menu driven program that reads a text file "BOOK.txt" and do the following on user's choice: 1. Display total number of uppercase alphabets in the file 2. Display how many times "the" word is present in the file 3. Display total number of lines starting with an uppercase vowel	<b>43</b>
<b>21</b>	A binary file "library.dat" stores the following details in the format given below:  <b>Book_id   Book_name   Publisher   Price</b>  Write a menu driven program to do the following: 1. Add book details for a new book to the file 2. Delete book details from the file, for a given book 3. Display the list of all the books 4. Display the details of a particular, where the book id is taken as input	<b>45</b>

22	<p>A binary file “phonebook.dat” stores name and phone number of users in following format:</p> <table border="0"> <tr> <td>Name</td><td>Phone</td></tr> <tr> <td>Jeevika</td><td>9876543212</td></tr> <tr> <td>Kriti</td><td>9856123444</td></tr> </table> <p>Write a menu driven program to do the following:</p> <ol style="list-style-type: none"> <li>1. Add a new record to the file</li> <li>2. Delete a record from the file</li> <li>3. Update the phone number of an existing name, where the name is taken as an input from the user. If there is no record for the input name, report error.</li> </ol>	Name	Phone	Jeevika	9876543212	Kriti	9856123444	50
Name	Phone							
Jeevika	9876543212							
Kriti	9856123444							
23	<p>Write a program to store the following details of sports’ performance (sport, competitions, prizes-won) of your school into a CSV file delimited with tab character. Write user defined functions for the following:</p> <ol style="list-style-type: none"> <li>1. To write data in the file.</li> <li>2. To read and display the details of sports where prizes-won is 0</li> </ol>	54						
24	<p>Write a program to store the following details of Log-in information (username, password, name) of a office into a CSV file. Write user defined functions for the following:</p> <ol style="list-style-type: none"> <li>1. To write data in the file</li> <li>2. To read and display data from the file</li> <li>3. To read data and write it to another CSV file with a different delimiter</li> </ol>	56						
25	<p>Create a module Number.py in python. The Number module has the following functions:</p> <ol style="list-style-type: none"> <li>1. prime(m,n) which takes m and n as argument and print all the prime numbers between m and n</li> <li>2. square(n) which takes n as argument and print the sum of all the numbers between 1 and n, which are divisible by 3 and 5.</li> <li>3. Find_num() which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 3000 (both included)</li> </ol> <p>Write a program which imports Number.py module and uses the different functions defined in the module.</p>	58						
26	<p>Create a module lengthconversion.py that stores functions for various length conversion:</p> <ol style="list-style-type: none"> <li>1. Miletokm() - to convert miles to kilometer</li> <li>2. Kmtomile() - to convert kilometer to miles</li> <li>3. Feettoinches – to convert feet to inches</li> <li>4. Inchestofeet – to convert inches to feet</li> </ol> <p>It should also store constant values such as value of(mile in kilometers and vice versa).</p> <p>[1 mile=1.609344 kilometers; 1 feet = 12 inches]</p> <p>Help() function should display proper information.</p> <p>Import the above module in a program and show the output for different conversion based on user input.</p>	60						
27	<p>Create a package Shape with the following modules:</p> <ol style="list-style-type: none"> <li>1. Rectangle.py – includes functions for calculating area and perimeter</li> <li>2. Square.py – includes functions for calculating area and perimeter</li> <li>3. Triangle.py – includes functions for calculating area and perimeter</li> <li>4. Circle.py – includes functions for calculating area and circumference</li> </ol> <p>Help() function should display proper information.</p> <p>Make sure that the above package meets the requirement of being a python package. Also, import above package and its modules using import command in two different programs with proper output statement.</p>	62						

<b>28</b>	Write a program to implement a list of names of countries as stack.	<b>64</b>
<b>29</b>	Write a program to implement a nested list(username, password) as stack.	<b>66</b>
<b>30</b>	SQL ASSIGNMENT - 1	<b>68</b>
<b>31</b>	SQL ASSIGNMENT – 2	<b>72</b>
<b>32</b>	<p>Write a program to connect to employee table in SQL and do the following on user choice:</p> <ol style="list-style-type: none"> <li>1. Fetch all records and display</li> <li>2. Delete a record from the table</li> <li>3. Insert a new record to the table</li> <li>4. Update a record value</li> </ol>	<b>75</b>

## Assignment 1

Write a program in python to input a start and stop value and do the following on user choice:

- i. Print all prime numbers in the range
- ii. Print all perfect numbers in the range
- iii. Print all special numbers in the range
- iv. Print all the palindromic number in the range

### OUTPUT:

```
1. Print all prime numbers in the range
2. Print all perfect numbers in the range
3. Print all special numbers in the range
4. Print all the palindromic number in the range
enter the choice: 1
Enter start range: 1
Enter stop range: 10
the prime numbers in the range are:
2
3
5
7
enter 'y' to continue : y
enter the choice: 2
Enter start range: 1
Enter stop range: 100
the perfect numbers in the range are:
6
28
enter 'y' to continue : y
enter the choice: 3
Enter start range: 1
Enter stop range: 200
the special numbers in the range are:
1
2
145
enter 'y' to continue : y
enter the choice: 4
Enter start range: 100
Enter stop range: 150
the palindromic numbers in the range are:
101
111
121
131
141
enter 'y' to continue : n
>>>
```

### CODE:

```
print("""1. Print all prime numbers in the range
2. Print all perfect numbers in the range
3. Print all special numbers in the range
4. Print all the palindromic number in the range""")
ch='y'
while (ch=='y' or ch=='Y'):
    n=int(input("enter the choice: "))
    start = int(input("Enter start range: "))
    stop = int(input("Enter stop range: "))
    #to find all prime numbers in the range
    #PRIME NUMBERS are numbers that have only 2 factors: 1 and themselves
    if n==1:
        print("the prime numbers in the range are:")
```

```

for num in range(start,stop + 1):
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                break
            else:
                print(num)
#to find all perfect numbers in the range
#PERFECT NUMBERS are positive integer that is equal to the sum of its proper divisors
elif n==2:
    print("the perfect numbers in the range are:")
    for num in range(start,stop + 1):
        Sum=0
        for i in range(1,num):
            if(num % i == 0):
                Sum = Sum + i
        if (Sum == num):
            print(num)
#to find all special numbers in the range
#SPECIAL NUMBERS are the ones whose sum of factorials of digits is equal to the
number itself
elif n==3:
    print("the special numbers in the range are:")
    for num in range(start,stop + 1):
        temp=num
        Sum=0
        while(temp>0):
            r=temp%10
            factorial=1
            for i in range(1,r + 1):
                factorial*=i
            Sum+=factorial
            temp//=10
        if(Sum == num):
            print(num)
elif n==4:
    print("the palindromic numbers in the range are:")
    for num in range(start,stop + 1):
        temp=num
        Sum=0
        while(temp>0):
            r=temp%10
            Sum=(Sum*10)+r
            temp=temp//10
        if(Sum == num):
            print(num)
ch=input("enter 'y' to continue : ")

```

**Assignment 2**

Write a program to input a tuple and print the sum of those values in the tuple, which are ending with 0 or 5.

**OUTPUT:**

```
Enter a tuple with integers(15,10,28,355)
Sum of all digit ending with 0 or 5 380
>>> |
```

**CODE:**

```
integer=eval(input("Enter a tuple with integers"))
sum5=0
for i in integer:
    if i%10==0 or i%10==5:
        sum5+=i
print("Sum of all digit ending with 0 or 5",sum5)
```

### Assignment 3

Write a program to input a tuple of integer elements and do the following on user choice:

- i. Print the reverse of each tuple element
- ii. Print the factorial of first digit of each tuple element
- iii. Print the maximum and minimum integer in the tuple

#### OUTPUT:

```
Enter tuple elements
```

```
(23,34,45,41,72)
```

- 1. Print the reverse of each tuple element
- 2. Print the factorial of first digit of each tuple element
- 3. Print the maximum and minimum integer in the tuple

```
Enter your choice1
```

```
Reverse of each tuple element
```

```
32
```

```
43
```

```
54
```

```
14
```

```
27
```

```
Do you want to continuey
```

- 1. Print the reverse of each tuple element
- 2. Print the factorial of first digit of each tuple element
- 3. Print the maximum and minimum integer in the tuple

```
Enter your choice2
```

```
Factorial of first digit of each tuple elemnt
```

```
23 2 2
```

```
34 3 6
```

```
45 4 24
```

```
41 4 24
```

```
72 7 5040
```

```
Do you want to continuey
```

- 1. Print the reverse of each tuple element
- 2. Print the factorial of first digit of each tuple element
- 3. Print the maximum and minimum integer in the tuple

```
Enter your choice3
```

```
Maximum value in the tuple 72
```

```
Minimum value in the tuple 23
```

```
Do you want to continueu
```

**CODE:**

```
def reverse(tup):
    print("Reverse of each tuple element\n")
    for i in tup:
        print(str(i)[::-1])

def factorial(tup):
    print("Factorial of first digit of each tuple elemnt")
    for i in tup:
        num=int(str(i)[0])
        fact=1
        for j in range(1,num+1):
            fact*=j
        print(i,num,fact,sep="   ")

Tuple=eval(input("\nEnter tuple elements\n"))
choice='y'
while choice=='y':
    print("1. Print the reverse of each tuple element")
    print("2. Print the factorial of first digit of each tuple element")
    print("3. Print the maximum and minimum integer in the tuple")
    select=int(input("Enter your choice"))
    if select==1:
        reverse(Tuple)
    elif select==2:
        factorial(Tuple)
    elif select==3:
        print("Maximum value in the tuple",max(Tuple))
        print("Minimum value in the tuple",min(Tuple))
    else:
        print("Wrong choice!!!")
    choice=input("Do you want to continue")
```

## Assignment 4

Write a program to input a list of integer elements and do the following on user choice:

- i. Print the sum of all the odd elements in the list
- ii. Remove duplicates from the list
- iii. Print the cube of all list elements, which are divisible by 2 and 3

### OUTPUT:

```
List : [11, 11, 2, 2, 2, 13, 24, 45]
Select your choice :
 1. Print the sum of all the odd elements in the list
 2. Remove duplicates from the list
 3. Print the cube of all list elements which are divisible by 2 and 3

Enter a choice:2
List after removing duplicates : [11, 2, 13, 24, 45]
Do you want to continuey

Enter a list:[11,11,2,2,2,13,24,45]
List : [11, 11, 2, 2, 2, 13, 24, 45]

Select your choice :
 1. Print the sum of all the odd elements in the list
 2. Remove duplicates from the list
 3. Print the cube of all list elements which are divisible by 2 and 3

Enter a choice:1
The sum of all the odd elements in the list : 80
Do you want to continuey

List : [11, 11, 2, 2, 2, 13, 24, 45]

Select your choice :
 1. Print the sum of all the odd elements in the list
 2. Remove duplicates from the list
 3. Print the cube of all list elements which are divisible by 2 and 3

Enter a choice:4
Not a valid input
Do you want to continuen

List : [11, 11, 2, 2, 2, 13, 24, 45]

Select your choice :
 1. Print the sum of all the odd elements in the list
 2. Remove duplicates from the list
 3. Print the cube of all list elements which are divisible by 2 and 3

Enter a choice:3
The cube of all list elements which are divisible by 2 and 3
13824
Do you want to continuey
```

**CODE:**

```
li=eval(input("Enter a list:"))
ch='y'
while ch=='y' or ch=='Y':
    print("List : ",li)
    print()

    print("Select your choice :
1. Print the sum of all the odd elements in the list
2. Remove duplicates from the list
3. Print the cube of all list elements which are divisible by 2 and 3")
    print()
    ent= int(input("Enter a choice:"))
    print()
    sum=0

    if ent==1:
        for i in li:
            if i%2!=0:
                sum=sum+i
        print("The sum of all the odd elements in the list : ", sum)
    elif ent==2:

        res = []
        for i in li:
            if i not in res:
                res.append(i)
        print("List after removing duplicates : ", res)

    elif ent==3:
        print("The cube of all list elements which are divisible by 2 and 3")

        for i in li:
            if i%2==0 and i%3==0:
                print("Cube of ",i," is ",i**3)
    else:
        print("Not a valid input")
    ch=input("Do you want to continue")
    print()
```

### Assignment 5

Write a menu driven program to input a matrix and do the following on user's choice

- i. Print sum of each row
- ii. Print sum of each column
- iii. Print all the diagonal elements

#### OUTPUT:

```
enter number of rows 3
enter number of columns 3
enter a number 1
enter a number 2
enter a number 3
enter a number 4
enter a number 5
enter a number 6
enter a number 7
enter a number 8
enter a number 9
1. Sum of each row
2. Sum of each column
3. Display diagnol elements
enter your choice 1
sum of rows:  6
sum of rows: 15
sum of rows: 24
```

```
enter number of rows 3
enter number of columns 3
enter a number 1
enter a number 2
enter a number 3
enter a number 4
enter a number 5
enter a number 6
enter a number 7
enter a number 8
enter a number 9
1. Sum of each row
2. Sum of each column
3. Display diagnol elements
enter your choice 2
sum of columns:  0
sum of columns: 0
sum of columns: 0
```

```

        enter number of rows 3
        enter number of columns 3
        enter a number 1
        enter a number 2
        enter a number 3
        enter a number 4
        enter a number 5
        enter a number 6
        enter a number 7
        enter a number 8
        enter a number 9
        1. Sum of each row
        2. Sum of each column
        3. Display diagnol elements
        enter your choice 3
        1
        3
        5
        5
        7
        9

```

**CODE:**

```

n=int(input('enter number of rows '))
m=int(input('enter number of columns '))
a=[[0]*m for i in range(n)]
for i in range(n):
    for j in range(m):
        a[i][j]=int(input('enter a number '))
sr=sc=ch=0
print("1. Sum of each row ")
print("2. Sum of each column ")
print("3. Display diagnol elements ")
ch=int(input("enter your choice "))
if (ch==1):
    for i in range(n):
        for j in range(m):
            sr+=a[i][j]
        print("sum of rows: ",sr)
        sr=0
elif (ch==2):

```

```
for i in range(m):
    for j in range(n):
        sc+=a[j][i]
    print("sum of columns: ",sc)
    sc=0
elif (ch==3):
    if (n==m):
        for i in range(m):
            for j in range(n):
                if (i==j):
                    print(a[i][j])
                if(i+j==n-1):
                    print(a[i][j])
else:
    print("wrong choice")
```

### Assignment 6

Write a menu driven program to input a matrix and do the following on user's choice:

- i. Print upper half of matrix
- ii. Print lower half of matrix
- iii. Print middle row and middle column elements

#### OUTPUT :

```
Enter number of rows3
```

```
: 1  
: 2  
: 3  
: 4  
: 5  
: 6  
: 7  
: 8  
: 9
```

```
Matrix is  
1 2 3  
4 5 6  
7 8 9
```

---

```
Matrix is  
1 2 3  
4 5 6  
7 8 9
```

- 1. Print upper half of matrix
- 2. Print lower half of matrix
- 3. Print middle row and middle column elements

```
Enter your choice
```

```
1  
Upper half elements  
1 2 3  
4 5  
7
```

```
Do you want to continuey
```

```
1. Print upper half of matrix
2. Print lower half of matrix
3. Print middle row and middle column elements
```

Enter your choice

2

Lower half elements

1

4 5

7 8 9

Do you want to continuey

```
1. Print upper half of matrix
2. Print lower half of matrix
3. Print middle row and middle column elements
```

Enter your choice

3

middle column elements

4

5

6

middle row elements

2

5

8

#### **CODE :**

```
def printmatrix():
    print("Matrix is")
    for j in range(r):
        for k in range(r):
            print(l[j][k],end=' ')
        print()
def uppermatrix():
    print("Upper half elements")
    co=r
    for i in range(r):
        for j in range(co):
            print(l[i][j],end=" ")
        co-=1
        print()
def lowermatrix():
    print("Lower half elements")
    for i in range(r):
        for j in range(i+1):
            print(l[i][j],end=" ")
        print()
```

```

def middle():
    print("middle column elements")
    for i in range(r):
        print(l[r//2][i])
    print("middle row elements")
    for i in range(r):
        print(l[i][r//2])

#Enter matrix
r=int(input("Enter number of rows"))
c=r
l=[0]*r
for j in range(r):
    l[j]=[0]*c
for i in range(r):
    for j in range(r):
        l[i][j]=int(input(": "))
printmatrix()
ch='y'
while ch=='y' or ch=='Y':
    print("1. Print upper half of matrix")
    print("2. Print lower half of matrix")
    print("3. Print middle row and middle column elements")
    choice=int(input("\nEnter your choice\n"))
    if choice==1:
        uppermatrix()
    elif choice==2:
        lowermatrix()
    elif choice==3:
        middle()
    else:
        print("Wrong choice")
    ch=input("Do you want to continue")

```

## Assignment 7

Write a menu driven program to input a matrix and do the following on user's choice:

1. Swap first row elements with last row elements
2. Swap first column elements with last column elements.

### OUTPUT:

```
Choose:  
1.To swap the first row elements with last  
row elements.  
2.To swap the first coloumn elements with l  
ast coloumn elements.  
Enter choice: 1  
Enter the no. of rows: 3  
Enter the no. of coloumns: 3  
Enter the element: 1  
Enter the element: 2  
Enter the element: 3  
Enter the element: 4  
Enter the element: 5  
Enter the element: 6  
Enter the element: 7  
Enter the element: 8  
Enter the element: 9  
Original matrix:  
1 2 3  
4 5 6  
7 8 9  
Changed matrix:  
7 8 9  
4 5 6  
1 2 3  
Do you wanna continue?
```

```
Do you wanna continue?  
yes  
Choose:  
1.To swap the first row elements with last  
row elements.  
2.To swap the first coloumn elements with l  
ast coloumn elements.  
Enter choice: 2  
Enter the no. of rows: 3  
Enter the no. of coloumns: 3  
Enter the element: 1  
Enter the element: 2  
Enter the element: 3  
Enter the element: 4  
Enter the element: 5  
Enter the element: 6  
Enter the element: 7  
Enter the element: 8  
Enter the element: 9  
Original matrix:  
1 2 3  
4 5 6  
7 8 9  
Changed matrix:  
3 2 1  
6 5 4  
9 8 7
```

```

Do you wanna continue?
yes
Choose:
1.To swap the first row elements with last
row elements.
2.To swap the first column elements with l
ast column elements.
Enter choice: 3
Invalid input.
Do you wanna continue?
no

#[QPython] Press enter to exit ...

```

**CODE:**

```

ch='yes'
while ch=='yes' or ch=='Yes':
    print("Choose:
1.To swap the first row elements with last row elements.
2.To swap the first column elements with last column elements.")
    n=int(input("Enter choice: "))
    if n==1:
        l=[]
        m = int(input("Enter the no. of rows: "))
        n = int(input("Enter the no. of columns: "))
        l=[0]*m
        for i in range(m):
            l[i]=[0]*n
        for i in range(m):
            for j in range(n):
                l[i][j]=int(input("Enter the element: "))
        print("Original matrix: ")
        for i in range(m):
            for j in range(n):
                print(l[i][j], end= ' ')
            print()
        print("Changed matrix: ")
        for j in range(m):
            t=l[0][j]
            l[0][j]=l[m-1][j]
            l[m-1][j]=t
        for i in range(m):
            for j in range(n):
                print(l[i][j], end= ' ')
            print()
    elif n==2:
        l=[]

```

```
m = int(input("Enter the no. of rows: "))
n = int(input("Enter the no. of columns: "))
l=[0]*m
for i in range(m):
    l[i]=[0]*n
for i in range(m):
    for j in range(n):
        l[i][j]=int(input("Enter the element: "))
print("Original matrix: ")
for i in range(m):
    for j in range(n):
        print(l[i][j], end= ' ')
    print()
print("Changed matrix: ")
for i in range(m):
    for j in range(n):
        print(l[i][j], end= ' ')
    print()
else:
    print("Invalid input.")
ch=input("Do you wanna continue?\n")
```

## Assignment 8

Write a program to input a square matrix. Print the sum of alternate elements.

For example :

Input matrix :      1 2 3  
                        4 5 6  
                        7 8 9

In the given matrix sum will be calculated for  $1 + 3 + 5 + 7 + 9$

Output :              Sum of alternate elements = 25

### OUTPUT:

enter row 3

enter column 4

enter 1

enter 2

enter 3

enter 4

enter 5

enter 6

enter 7

enter 8

enter 9

enter 1

enter 2

enter 3

1 2 3 4

5 6 7 8

9 1 2 3

Sum of alternate elements 27

```
enter row 3
enter column 3
enter 1
enter 2
enter 3
enter 4
enter 5
enter 6
enter 7
enter 8
enter 9
1 2 3
4 5 6
7 8 9
Sum of alternate elements 25
```

**CODE:**

```
r=int(input('enter row '))
c=int(input('enter column '))
l=[0]*r
for j in range(r):
    l[j]=[0]*c
for i in range(r):
    for k in range(c):
        l[i][k]=int(input('enter '))
for i in range(r):
    for k in range(c):
        print(l[i][k],end=' ')
    print()
```

#SUM OF ALTERNATE ELEMENTS

```
s=0
count=1
for i in range(r):
    for k in range(c):
        if count%2!=0:
            s+=l[i][k]
        count+=1

print ("Sum of alternate elements",s)
```

### **Assignment 9**

Write a Python program to input two sorted list n ascending order and merge them to form a new list in ascending order.

#### **OUTPUT:**

```
Enter first list: [2,17,9,3,10]
Enter second list: [23,56,9,11]
New merged list in ascending order:
[2, 3, 9, 9, 10, 11, 17, 23, 56]
```

```
Enter first list: [-1,-9,-8,-3]
Enter second list: [2,5,19,1,4]
New merged list in ascending order:
[-9, -8, -3, -1, 1, 2, 4, 5, 19]
```

#### **CODE:**

```
la=eval(input("Enter first list: "))
lb=eval(input("Enter second list: "))
la.sort()
lb.sort()
lc=la+lb
lc.sort()
print("New merged list in ascending order: ",lc)
```

## Assignment 10

Write a Python program to input two sorted list n descending order and merge them to form a new list in ascending order.

### OUTPUT:

```
enter numbers56,27,82
enter number23,4,53
[23, 4, 53]
ipdb>
```

### CODE:

```
I = eval(input("enter number"))
li = eval(input("enter number"))
i = 0
j = 0
I_len = len(I)
li_len = len(li)
n = []
while (i < I_len and j < li_len):
    if (I[i] < li[j]):
        n.append(I[i])
        i = i + 1
    else:
        n.append(li[j])
        j = j + 1
if (i > I_len):
    while (i < I_len):
        n.append(I[i])
        i = i + 1
if (j > li_len):
    while (j < li_len):
        n.append(li[j])
        j = j + 1
print(n)
```

### **Assignment 11**

Write a program to input a tuple and sort it based on user choice:

1. Using Bubble sort
2. Using insertion sort

#### **OUTPUT:**

```
Enter a list of elements  
[23,67,8,9,12,26,18]  
  
Choose a sorting method  
  
1. Bubble sorting  
2. Selection sorting  
  
Enter your choice1  
Sorted list  
[8, 9, 12, 18, 23, 26, 67]  
  
Do you want to continue? (y/n)y
```

```
Enter a list of elements  
[12,-1,-10,56,34,22,101]  
  
Choose a sorting method  
  
1. Bubble sorting  
2. Selection sorting  
  
Enter your choice2  
[-10, -1, 12, 22, 34, 56, 101]  
  
Do you want to continue? (y/n)y
```

```
Enter a list of elements  
[12,23,34,56]  
  
Choose a sorting method  
  
1. Bubble sorting  
2. Selection sorting  
  
Enter your choice3  
Wrong choice. Try again !!!!  
  
Do you want to continue? (y/n)n
```

**CODE:**

```
def bubbleSort(arr):
    n = len(arr)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

def selectionSort(A):
    for i in range(len(A)):
        min_idx = i
        for j in range(i+1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j
        A[i], A[min_idx] = A[min_idx], A[i]
cont='y'
while cont=='y':
    arr=eval(input("\nEnter a list of elements\n"))
    print("\nChoose a sorting method")
    print("\n1. Bubble sorting")
    print("2. Selection sorting\n")
    ch=int(input("Enter your choice"))
    if ch==1:
        bubbleSort(arr)
        print("Sorted list\n",arr)
    elif ch==2:
        selectionSort(arr)
        print(arr)
    else:
        print("Wrong choice. Try again !!!!")
    cont=input("Do you want to continue? (y/n)").lower()
```

## Assignment 12

Write a program to input a string and do the following on user's choice:

1. Print all four letter words
2. Print all palindromic words
3. Print all the words starting with 'a' or 'A'
4. Print all the words which starts and ends with a vowel

### OUTPUT:

```
Enter a string
```

```
A palindrome refer to a word , sentence , verse , or even number that  
reads the same backward and forward . James Joyce coined the longest  
palindrome in English tattarrattat in his 1922 Ulysses to imitate the  
sound of a knock on the door .
```

1. Print all four letter words
2. Print all palindromic words
3. Print all the words starting with 'a' or 'A'
4. Print all the words which starts and ends with a vowel

```
Enter your choice
```

```
1
```

```
All four letter words are
```

```
word  
even  
that  
same  
1922  
door
```

```
Do you want to continue? (y/n)y
```

1. Print all four letter words
2. Print all palindromic words
3. Print all the words starting with 'a' or 'A'
4. Print all the words which starts and ends with a vowel

```
Enter your choice
```

```
2
```

```
All palindrome are
```

```
refer  
tattarrattat
```

```
Do you want to continue? (y/n)y
```

1. Print all four letter words
2. Print all palindromic words
3. Print all the words starting with 'a' or 'A'
4. Print all the words which starts and ends with a vowel

Enter your choice

3

All words starting with a or A

A  
a  
and  
a

Do you want to continue? (y/n)y

1. Print all four letter words
2. Print all palindromic words
3. Print all the words starting with 'a' or 'A'
4. Print all the words which starts and ends with a vowel

Enter your choice

4

All words starting and ending with a vowel

imitate

Do you want to continue? (y/n)y

1. Print all four letter words
2. Print all palindromic words
3. Print all the words starting with 'a' or 'A'
4. Print all the words which starts and ends with a vowel

Enter your choice

5

Wrong choice!!!!

Do you want to continue? (y/n)n

**CODE:**

```
def fourLetterWord(st):  
    words=st.split()  
    print("\nAll four letter words are\n")  
    for word in words:
```

```

if len(word)==4:
    print(word)

def palindrome(st):
    words=st.split()
    print("\nAll palindrome are\n")
    for word in words:
        if len(word)>1:
            if word==word[::-1]:
                print(word)

def wordWithA(st):
    words=st.split()
    print("\nAll words starting with a or A\n")
    for word in words:
        if word[0].lower()=='a':
            print(word)

def vowel(st):
    words=st.split()
    print("\nAll words starting and ending with a vowel\n")
    l=['a','e','i','o','u']
    for word in words:
        if len(word)>1:
            if word[0].lower() in l and word[-1].lower() in l:
                print(word)

cont='y'
st=input("\nEnter a string\n")
while cont=='y':
    print("1. Print all four letter words")
    print("2. Print all palindromic words")
    print("3. Print all the words starting with 'a' or 'A'")
    print("4. Print all the words which starts and ends with a vowel")
    ch=int(input("\nEnter your choice\n"))
    if ch==1:
        fourLetterWord(st)
    elif ch==2:
        palindrome(st)
    elif ch==3:
        wordWithA(st)
    elif ch==4:
        vowel(st)
    else:
        print("Wrong choice!!!!")
    cont=input("Do you want to continue? (y/n)").lower()

```

### Assignment 13

Write a program to find the number of vowels, consonants, digits and white space characters in a string.

#### OUTPUT:

```
enter string:  
A journey of 1000 miles begin with a single step  
total number of vowels= 14  
total number of consonants= 21  
spaces= 9  
digits= 4
```

```
enter string:  
Adversity and loss makes a man wise.  
total number of vowels= 11  
total number of consonants= 18  
spaces= 6  
digits= 0
```

#### CODE:

```
str1=input("enter string:\n")  
v=c=sp=d=0  
vowel=['a','e','i','o','u','A','E','I','O','U']  
for i in str1:  
    if i.isalpha()==True:  
        if i in vowel:  
            v+=1  
        else:  
            c+=1  
    if i.isspace()==True:  
        sp+=1  
    if i.isdigit()==True:  
        d+=1  
print ( "total number of vowels=",v)  
print ( "total number of consonants=",c)  
print("spaces=",sp)  
print("digits=",d)
```

## **Assignment 14**

Write a Python program to check the validity of password input by users.

Validation :

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [\$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

### **OUTPUT:**

```
Enter your password  
SM@2
```

Take a password which has 6 to 16 characters

```
Enter your password  
Sm21345
```

Not a strong password

You must have one digit(0-9),one character(\$,#,@) and one uppercase and lowercase charcater

```
Enter your password  
SM@2020a
```

You have a strong password

### **CODE:**

```
password=input("\nEnter your password\n")  
flag=False  
num=("1","2","3","4","5","6","7","8","9","0")  
sp=("$","#","@")  
ch_sp=ch_num=ch_a=ch_A=0  
if len(password)>=6 and len(password)<=16:  
    for i in password:  
        if i in sp:
```

```
ch_sp+=1
elif i in num:
    ch_num+=1
elif i>='a' and i<='z':
    ch_a+=1
elif i>='A' and i<='Z':
    ch_A+=1
if ch_sp>0 and ch_num>0 and ch_a>0 and ch_A>0:
    flag=True
if flag==True:
    print("\nYou have a strong password\n")
else:
    print("\nNot a strong password\n")
    print("\nYou must have one digit(0-9),one character($,#,@) and one uppercase and lowercase charcater" )
else:
    print("\nTake a password which has 6 to 16 characters")
```

## Assignment 15

Write a program to check if two strings are anagram or not.

NOTE: An anagram of a string is another string which contains all the characters of original string in different sequence. Example "abcd" and "dcba" are anagram.

### OUTPUT:

```
Enter first string  
silent  
  
Enter second string  
listen  
  
The strings are anagrams.
```

```
Enter first string  
Triangle  
  
Enter second string  
Integral  
  
The strings are anagrams.
```

```
Enter first string  
hello  
  
Enter second string  
hell  
  
The strings aren't anagrams.  
'
```

### Code:

```
# function to check if two strings are  
# anagram or not  
def check(s1, s2):  
  
    # the sorted strings are checked  
    if(sorted(s1.lower())== sorted(s2.lower())):  
        print("\nThe strings are anagrams.")  
    else:  
        print("\nThe strings aren't anagrams.")  
  
# driver code  
s1 =input("\nEnter first string\n")  
s2 =input("\nEnter second string\n")  
check(s1, s2)
```

## **Assignment 16**

Write a program that repeatedly asks the user to enter product name and price. Store all of these in a dictionary whose keys are Product name and value is the price. From the entered dictionary print average price and product name which has the highest price.

### **OUTPUT:**

```
Enter product nameSoap
Enter price of the product10
Do you wish to add more product names and prices? Y/NY
Enter product nameChocolate
Enter price of the product20
Do you wish to add more product names and prices? Y/NY
Enter product namePencil
Enter price of the product30
Do you wish to add more product names and prices? Y/NY
Enter product nameShoes
Enter price of the product1000
Do you wish to add more product names and prices? Y/NN
{'Soap': 10, 'Chocolate': 20, 'Pencil': 30, 'Shoes': 1000}
Product with maximum price Shoes
Average price 265.0
```

### **Code-**

```
d={}
name=[]
price=[]
s=0
c='y'
while c=='y' or c=='Y':
    a=input("Enter product name")
    b=int(input("Enter price of the product"))
    name.append(a)
    price.append(b)
    s+=b
    c=input("Do you wish to add more product names and prices? Y/N")
    if c=='n' or c=='N':
        break
x=len(name)
for i in range(x):
    d[name[i]]=price[i]
print(d)

for i in d:
    if(d[i]==max(price)):
        print("Product with maximum price",i)
print("Average price",s/x)
```

## Assignment 17

Create a dictionary whose keys are month names and whose values are the number of days in the corresponding months.

1. Ask the user to enter a month name and use the dictionary to tell how many days are there in the month.
2. Print all the keys in alphabetical order.
3. Print names of all the months with 31 days

### OUTPUT:

```
1. Ask the user to enter a month name and use the dictionary to tell how many days are there in the month.
2. Print all the keys in alphabetical order.
3. Print names of all the months with 31 days
Enter your choice:1
Enter a month name:November
November has 30 days in a month
Enter Y if you want to continue:Y
Enter your choice:2
The month names in alphabetical order ['April', 'August', 'December', 'February', 'January', 'July', 'June', 'March', 'May', 'November', 'October', 'September']
Enter Y if you want to continue:Y
Enter your choice:3
The months which have 31 days are:
January
March
May
July
August
October
December
Enter Y if you want to continue:N
```

### CODE:

```
d={'January':31,'February':28,'March':31,'April':30,'May':31,'June':30,'July':31,'August':31
,'September':30,'October':31,'November':30 , 'December':31 }

print("1. Ask the user to enter a month name and use the dictionary to tell how many days are there in the month.

2. Print all the keys in alphabetical order.

3. Print names of all the months with 31 days")

ch='Y'

while ch=='Y':

    ch=int(input('Enter your choice:'))

    if ch==1:

        a=input('Enter a month name:')

        for i in d:

            if a==i or a.capitalize()==i:

                print(a,'has',d[i],'days in a month')
```

```
elif ch==2:  
    l=[]  
    for i in d:  
        l.append(i)  
    l.sort()  
    print('The month names in alphabetical order',l)  
  
elif ch==3:  
    print('The months which have 31 days are:')  
    for i in d:  
        if d[i]==31:  
            print(i)  
  
else:  
    print('Invalid choice')  
ch=input('Enter Y if you want to continue:')
```

## **Assignment 18**

Write a program to create a text file "Diary.txt" and do the following on user choice:

1. Display the number of vowels
2. Display the number of words
3. Display the number of lines
4. Display the most common word in file

### **OUTPUT:**

```
1. Display the number of vowels
2. Display the number of words
3. Display the number of lines
4. Display the most common word in file
enter choice1
The number of vowels are 10
Do you wan to continu? Y/Ny
enter choice2
The number of words are 8
Do you wan to continu? Y/Ny
enter choice3
The number of lines are 2
Do you wan to continu? Y/Ny
enter choice4
The most repeated word is hi
Do you wan to continu? Y/Nn
>>>
```

### **CODE:**

```
f=open('diary.txt','w')
line='hello hi hi \n'
line1='hi how are you\n'
f.write(line)
f.write(line1)
f.close()
vowels=0
count=0
lines=0
word=[]
cont='y'
f=open('diary.txt','r')
vowel=['A','E','I','O','U','a','e','i','o','u']
rea=f.readlines()
print("1. Display the number of vowels")
print("2. Display the number of words")
print("3. Display the number of lines")
print("4. Display the most common word in file")
while cont=='y' or cont=='Y':
    ch=int(input("enter choice"))
```

```

if ch==1:
    for a in rea:
        for i in a:
            for j in range(10):
                if i==vowel[j]:
                    vowels=vowels+1
    print("The number of vowels are",vowels)
    cont=input("Do you want to continue? Y/N")
elif ch==2:
    for a in rea:
        words=a.split(" ")
        for i in words:
            count=count+1
    print("The number of words are",count)
    cont=input("Do you want to continue? Y/N")
elif ch==3:
    for a in rea:
        lines=lines+1
    print("The number of lines are",lines)
    cont=input("Do you want to continue? Y/N")
elif ch==4:
    for a in rea:
        words=a.split()
        for i in words:
            word.append(i)
    co=words.count(words[0])
    nax=co
    for b in word:
        cou = word.count(b)
        if cou>co:
            nax=b
    print("The most repeated word is",nax)
    cont=input("Do you want to continue? Y/N")
else:
    print("Invalid Choice")
    cont=input("Do you want to continue? Y/N")
f.close()

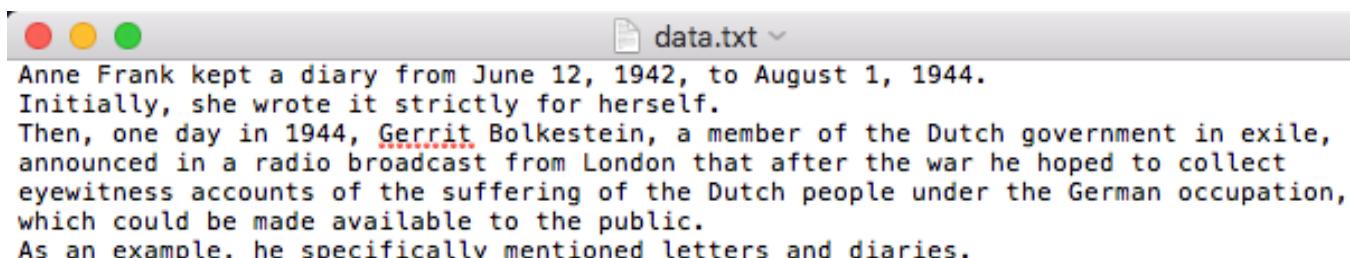
```

### **Assignment 19**

Write a menu driven program that reads a text file “data.txt” and do the following on user’s choice:

1. Display total number of uppercase alphabets in the file
2. Display how many times “the” word is present in the file
3. Display total number of lines starting with an uppercase vowel

#### **OUTPUT:**



The screenshot shows a terminal window with a title bar "data.txt". The window contains the following text:  
Anne Frank kept a diary from June 12, 1942, to August 1, 1944.  
Initially, she wrote it strictly for herself.  
Then, one day in 1944, Gerrit Bolkestein, a member of the Dutch government in exile,  
announced in a radio broadcast from London that after the war he hoped to collect  
eyewitness accounts of the suffering of the Dutch people under the German occupation,  
which could be made available to the public.  
As an example, he specifically mentioned letters and diaries.

1. Display total number of uppercase alphabets in the file
2. Display how many times “the” word is present in the file
3. Display total number of lines starting with an uppercase vowel

Enter your choice

1

Total uppercase alphabets 13

Do you want to continue? (y/n)y

1. Display total number of uppercase alphabets in the file
2. Display how many times “the” word is present in the file
3. Display total number of lines starting with an uppercase vowel

Enter your choice

2

Total <the> words in the file 6

Do you want to continue? (y/n)y

1. Display total number of uppercase alphabets in the file
2. Display how many times "the" word is present in the file
3. Display total number of lines starting with an uppercase vowel

Enter your choice

3

Total number of lines starting with uppercase vowel 3

Do you want to continue? (y/n)y

1. Display total number of uppercase alphabets in the file
2. Display how many times "the" word is present in the file
3. Display total number of lines starting with an uppercase vowel

Enter your choice

4

Wrong choice. Try again!!!

Do you want to continue? (y/n)n

**CODE:**

```
def readcharacters():
    f = open("data.txt", 'r')
    count=0
    st=f.read()
    for i in st:
        if i>='A' and i<='Z':
            count+=1
    f.close()
    return count
```

```
def thewords():
    f = open("data.txt", 'r')
    count=0
    st=f.read()
    words=st.split()
    for i in words:
        if i.lower() == 'the':
            count+=1
    f.close()
    return count
```

```
def readLines():
    f = open("data.txt", 'r')
```

```

st=f.readlines()
count=0
uppervowel=['A','E','I','O','U']
for i in st:
    if i[0] in uppervowel:
        count+=1
f.close()
return count

cont='y'
while cont.lower()=='y':
    print("\n1. Display total number of uppercase alphabets in the file
          \n2. Display how many times "the" word is present in the file
          \n3. Display total number of lines starting with an uppercase vowel")
ch=int(input("Enter your choice\n"))
if ch==1:
    print("Total uppercase alphabets",readcharacters())
elif ch==2:
    print("Total <the> words in the file",thewords())
elif ch==3:
    print("Total number of lines starting with uppercase vowel",readLines())
else:
    print("Wrong choice. Try again!!!")
cont=input("Do you want to continue? (y/n)")

```

## Assignment 20

Write a menu driven program that reads a text file "BOOK.txt" and do the following on user's choice:

1. Display total number of uppercase alphabets in the file
2. Display how many times "the" word is present in the file
3. Display total number of lines starting with an uppercase vowel

### OUTPUT:

```
1.Display total number of uppercase alphabets in the file.
2.Display how many time "the" is present in the file.
3.Display total number of lines starting with an uppercase vowel.
Enter your choice
1
The total number of uppercase alphabets in the text file ar
e= 6
enter y to continue
y
Enter your choice
2
The total numeber of times "the" is present in the file is
5
enter y to continue
y
Enter your choice
3
Total number of lines strating with an uppercase vowel are=
1
enter y to continue
■
```

### CODE:

```
myfile=open('Book.txt','w')
myfile.write('This is a text file.\n')
myfile.write('The program demands me to write the word the so I can check the number of
times I write the word the\n')
myfile.write('lol\n')
myfile.write('Oh, I also have to start my sentence with a vowel/n')
myfile.close()
print('1.Display total number of uppercase alphabets in the file.')
print('2.Display how many time "the" is present in the file.')
print('3.Display total number of lines starting with an uppercase vowel.')
ch='y'
while(ch=='y' or ch=='Y'):
    a=int(input('Enter your choice\n'))
    if(a==1):
        myfile=open('Book.txt')
```

```
text=myfile.read()
count=0
for i in text:
    if i.isupper():
        count+=1
print('The total number of uppercase alphabets in the text file are=',count)
myfile.close()
elif(a==2):
    myfile=open('Book.txt')
    texts=myfile.readlines()
    counts=0
    for i in texts:
        for j in i.split(' '):
            if j.lower()=='the':
                counts+=1
    print('The total number of times "the" is present in the file is',counts)
    myfile.close()
elif(a==3):
    myfile=open('Book.txt')
    t=myfile.readlines()
    c=0
    vowel=['A','E','I','O','U']
    for i in t:
        if i[0] in vowel:
            c+=1
    print('Total number of lines starting with an uppercase vowel are=',c)
else:
    ('invalid choice')
ch=input('enter y to continue\n')
```

## **Assignment 21**

A binary file "library.dat" stores the following details in the format given below:

**Book\_id   Book\_name   Publisher   Price**

Write a menu driven program to do the following:

1. Add book details for a new book to the file
2. Delete book details from the file, for a given book
3. Display the list of all the books
4. Display the details of a particular, where the book id is taken as input

### **OUTPUT:**

```
Choices:  
1. Add book details for a new book to the file  
2. Delete book details from the file for a given book  
3. Display the list of all the books  
4. Display the details of a particular book, where the book id is taken as input  
enter your choice1  
enter book id for the new bookabl2  
enter the name of the new bookMonsoon  
enter the name of the publisher of the new bookPC  
enter the price of the new book340  
details of the new book have been added to the file  
do you want to continue?(y/n)y  
Choices:  
1. Add book details for a new book to the file  
2. Delete book details from the file for a given book  
3. Display the list of all the books  
4. Display the details of a particular book, where the book id is taken as input  
enter your choice1  
enter book id for the new bookcd34  
enter the name of the new bookSummer  
enter the name of the publisher of the new bookQU  
enter the price of the new book220  
details of the new book have been added to the file  
do you want to continue?(y/n)y  
Choices:  
1. Add book details for a new book to the file  
2. Delete book details from the file for a given book  
3. Display the list of all the books  
4. Display the details of a particular book, where the book id is taken as input  
enter your choice1  
enter book id for the new bookef56  
enter the name of the new bookWinter  
enter the name of the publisher of the new bookSZ  
enter the price of the new book450  
details of the new book have been added to the file  
do you want to continue?(y/n)y  
Choices:  
1. Add book details for a new book to the file  
2. Delete book details from the file for a given book  
3. Display the list of all the books  
4. Display the details of a particular book, where the book id is taken as input
```

```
    . . .
Choices:
1. Add book details for a new book to the file
2. Delete book details from the file for a given book
3. Display the list of all the books
4. Display the details of a particular book, where the book id is taken as input
enter your choice3
details of all the books:

Book 1 :
Book_id - abl2
Book_name - Monsoon
Publisher - PC
Price - 340
Book 2 :
Book_id - cd34
Book_name - Summer
Publisher - QU
Price - 220
Book 3 :
Book_id - ef56
Book_name - Winter
Publisher - SZ
Price - 450
do you want to continue?(y/n)y
Choices:
1. Add book details for a new book to the file
2. Delete book details from the file for a given book
3. Display the list of all the books
4. Display the details of a particular book, where the book id is taken as input
enter your choice4
enter book id of the book whose details are to be displayed:cd34
details of book with id cd34 are as follows:

Book_id - cd34
Book_name - Summer
Publisher - QU
Price - 220
do you want to continue?(y/n)y
. . .
```

```
Choices:
1. Add book details for a new book to the file
2. Delete book details from the file for a given book
3. Display the list of all the books
4. Display the details of a particular book, where the book id is taken as input
enter your choice2
enter the name of the book whose details are to be deletedSummer
details of the book Summer have been deleted
do you want to continue?(y/n)y
Choices:
1. Add book details for a new book to the file
2. Delete book details from the file for a given book
3. Display the list of all the books
4. Display the details of a particular book, where the book id is taken as input
enter your choice3
details of all the books:

Book 1 :
Book_id - ab12
Book_name - Monsoon
Publisher - PC
Price - 340
Book 2 :
Book_id - ef56
Book_name - Winter
Publisher - SZ
Price - 450
do you want to continue?(y/n)n
>>> |
```

### CODE:

```
import pickle

def adddata():
    f=open('library.dat.txt','ab')
    d={}
    d['Book_id']=input('enter book id for the new book')
    d['Book_name']=input('enter the name of the new book')
    d['Publisher']=input('enter the name of the publisher of the new book')
    d['Price']=input('enter the price of the new book')
    pickle.dump(d,f)
    print('details of the new book have been added to the file')
    f.close()
```

```

def deldata():
    f=open('library.dat.txt','rb+')
    fo=open('extra.dat.txt','wb')
    e=input('enter the name of the book whose details are to be deleted')
    t=0
    try:
        while True:
            x=pickle.load(f)
            if x['Book_name']!=e:
                pickle.dump(x,fo)
            else:
                print('details of the book',e,'have been deleted')
                t+=1
    except EOFError:
        f.close()
        fo.close()
        if t==0:
            print('Book not found therefore cannot be deleted')
    fr=open('library.dat.txt','wb')
    fro=open('extra.dat.txt','rb+')
    try:
        while True:
            z=pickle.load(fro)
            pickle.dump(z,fr)
    except EOFError:
        fr.close()
        fro.close()

def rwhole():
    f=open('library.dat.txt','rb')
    print('details of all the books:\n')
    try:
        s=1
        while True:
            y=pickle.load(f)
            print('Book',s,':')
            for i in y:
                print(i,'-',y[i])
            s+=1
    except EOFError:
        if s==1:
            print('file is empty')
        f.close()

def rsingle():
    f=open('library.dat.txt','rb')
    k=input('enter book id of the book whose details are to be displayed:')
    t=0

```

```

try:
    while True:
        p=pickle.load(f)
        if p['Book_id']==k:
            print('details of book with id',k,'are as follows:\n')
            for i in p:
                print(i,'-',p[i])
            t+=1
    except EOFError:
        if t==0:
            print('Book not in file')
        f.close()

g='y'
while g=='y':
    print('Choices:')
    print('1. Add book details for a new book to the file')
    print('2. Delete book details from the file for a given book')
    print('3. Display the list of all the books')
    print('4. Display the details of a particular book, where the book id is taken as input')
    a=int(input('enter your choice'))
    if(a==1):
        adddata()
    elif(a==2):
        deldata()
    elif(a==3):
        rwhole()
    elif(a==4):
        rsingle()
    else:
        print('choice not available, try again if you want')
    g=input('do you want to continue?(y/n)')

```

## Assignment 22

A binary file "phonebook.dat" stores name and phone number of users in following format:

Name	Phone
Jeevika	9876543212
Kriti	9856123444

Write a menu driven program to do the following:

1. Add a new record to the file
2. Delete a record from the file
3. Update the phone number of an existing name, where the name is taken as an input from the user. If there is no record for the input name, report error.

### OUTPUT:

```
MacBook-Pro:python projects yug$ /Library/Frameworks/Python.framework/Versions/3.7/bin/python3 "/Users/yug/Documents/python projects/Practile_file.py"
1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice3
Name Phone
yug 1000
Do you want to continuey
1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice1
Enter name to be added: Maahi
Enter number: 123456789
Do you want to add more records: n
Do you want to continuey
1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice3
Name Phone
yug 1000
Maahi 123456789
Do you want to continuey
1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice2
Enter the name whose record is to be updated: Maahi
Enter new phone number: 111111
Record updated
Do you want to continuey
1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice3
Name Phone
yug 1000
Maahi 111111
Do you want to continuey
```

```

1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice4
Enter the name whose record has to be deleted: yug
Record deleted
Do you want to continuey
1. Add data to a file
2. Update file data
3. Read data using try
4. Delete data from the file
Enter your choice3
Name Phone
Maahi 111111
Do you want to continue

```

```

import pickle

#add more data
def Add_data():
    fw = open("phonebook.dat","ab+")
    ch = 'y'
    phonebook={}
    while ch == 'y' :
        name = input("Enter name to be added: ")
        num = int(input("Enter number: "))
        phonebook['name'] = name
        phonebook['Phone'] = num
        pickle.dump(phonebook, fw)
        ch = input("Do you want to add more records: ")
    fw.close()

#update record
def update_data():
    fw=open('phonebook.dat','rb+')
    e={}
    name = input("Enter the name whose record is to be updated: ")
    found=False
    try:
        while True:
            rpos=fw.tell()
            e=pickle.load(fw)
            if e['name'] == name:
                phone = int(input('Enter new phone number: '))
                e['Phone'] = phone
                fw.seek(rpos)
                pickle.dump(e,fw)
                found=True
    except EOFError:
        if found==False:

```

```

        print("Record not found")
    else:
        print("Record updated")
    fw.close()

#delete data
def delete_data():
    fw = open("phonebook.dat","rb")
    fr = open("temp.dat", "wb+")
    e = {}
    found = False
    Name = input('Enter the name whose record has to be deleted: ')
    try:
        while True:
            e = pickle.load(fw)
            if e['name'] != Name:
                pickle.dump(e, fr)
                found = True

    except EOFError:
        if found == True:
            print("Record deleted")
        else:
            print("Record not deleted")
        fr.close()
        fw.close()

```

```

#read data
def read_data():
    fw = open("phonebook.dat","rb")
    e = {}
    print("Name Phone")
    try:
        while True:
            e = pickle.load(fw)
            print(e['name'],e['Phone'])
    except EOFError:
        fw.close()

```

```

#provides choices to user
ch='y'
while ch=='Y' or ch=='y':
    print("1. Add data to a file")
    print("2. Update file data")
    print("3. Read data using try")

```

```
print("4. Delete data from the file")
c=int(input("Enter your choice"))
if c==1:
    Add_data()
elif c==2:
    update_data()
elif c==3:
    read_data()
elif c==4:
    delete_data()
    fw = open("temp.dat","rb")
    fr = open("phonebook.dat","wb+")
    e = {}
    try:
        while True:
            e = pickle.load(fw)
            pickle.dump(e, fr)
    except EOFError:
        fw.close()
        fr.close()
else:
    print("Wrong choice")
ch=input("Do you want to continue")
```

### **Assignment 23**

Write a program to store the following details of sports' performance (sport, competitions, prizes-won) of your school into a CSV file delimited with tab character. Write user defined functions for the following:

1. To write data in the file.
2. To read and display the details of sports where prizes-won is 0.

### **OUTPUT:**

```
Enter the number of entries you want to enter:  
4  
Enter the name of the sport, number of competitions participated in and number of prizes won  
chess  
4  
5  
Enter the name of the sport, number of competitions participated in and number of prizes won  
basketball  
17  
0  
Enter the name of the sport, number of competitions participated in and number of prizes won  
football  
14  
9  
Enter the name of the sport, number of competitions participated in and number of prizes won  
squash  
3  
0  
Sport: basketball  
Number of competitions participated in: 17  
There are no prizes won in this sport.  
Sport: squash  
Number of competitions participated in: 3  
There are no prizes won in this sport.  
PS D:\Python Programs> []
```

### **CODE:**

```
import csv  
  
def readCsvFile(openedFile):  
    reader = csv.reader(openedFile)  
    skipOnce = 0 # this will help us skip the header values in the tsv file.  
    for k in reader:  
        if skipOnce > 0:  
            i = k[0].split('\t')
```

```

if int(i[2]) == 0:
    print("\nSport: ", i[0], sep="")
    print("Number of competitions participated in: ", i[1], sep="")
    print("There are no prizes won in this sport.", end = '\n')
else:
    skipOnce += 1

def writeCsvFile(openedFile):
    writer = csv.writer(openedFile, delimiter='\t')
    print("Enter the number of entries you want to enter:\n")
    number = int(input())

    i = 0
    while i < number:
        print("Enter the name of the sport, number of competitions participated in and number of prizes won\n")
        sportName = input()
        competitions = int(input())
        prizesWon = int(input())
        writer.writerow([sportName, int(competitions), int(prizesWon)])
        i += 1

def initializeFile(openedFile):
    firstRow = ['Sports', 'Competitions', 'Prizes Won']
    csv.writer(openedFile, delimiter = '\t').writerow(firstRow)

with open('Sports.csv', 'w+', newline='\n') as csvFile:
    initializeFile(csvFile)
    writeCsvFile(csvFile)
    csvFile.seek(0)
    readCsvFile(csvFile)

```

## Assignment 24

Write a program to store the following details of Log-in information (username, password, name) of a office into a CSV file. Write user defined functions for the following:

1. To write data in the file
2. To read and display data from the file
3. To read data and write it to another CSV file with a different delimiter

### OUTPUT:

```
WRITING LOG IN INFORMATION IN A CSV FILE
Enter a filename:office.csv
Enter Name:Ananya Sharma
Enter Username:ananya.sharma
Enter Password:1234

Do you want to continue?: y/ny

Enter Name:Raunak Ahuja
Enter Username:raunak_ahuja
Enter Password:5647

Do you want to continue?: y/nn

READING AND DISPLAYING THE DATA FROM CSV FILE
['Name', 'Username', 'Password']
['Ananya Sharma', 'ananya.sharma', '1234']
['Raunak Ahuja', 'raunak_ahuja', '5647']

READING DATA AND WRITING IT INTO ANOTHER FILE
Enter the other filename:office2.csv
Data written in another file

Reading data from office2.csv
['Name:Username:Password']
['Ananya Sharma:ananya.sharma:1234']
['Raunak Ahuja:raunak_ahuja:5647']
>>>
```

### CODE:

```
#write data in file
def writedata(filename):
    import csv
    fopen=open(filename,"a")
    fwrite=csv.writer(fopen)
    fwrite.writerow(['Name','Username','Password'])
    ch='y'
    while ch=='y':
        Name=input('Enter Name:')
        Username=input('Enter Username:')
        Password=input('Enter Password:')
        rec=[Name,Username,Password]
        fwrite.writerow(rec)
        print()
        ch=input('Do you want to continue?: y/n')
```

```

print()
fopen.close()

#read and display data
def displaydata(filename):
    import csv
    with open(filename,"r",newline="\r\n") as fh:
        creader=csv.reader(fh)
        for rec in creader:
            print(rec)

#read and write in another csv file
def writefile(filename,filename2):
    import csv
    fopen=open(filename2,'a')
    fwrite=csv.writer(fopen, delimiter=':')
    with open(filename,"r",newline="\r\n") as fh:
        abc=csv.reader(fh)
        for rec in abc:
            a=rec
            fwrite.writerow(a)
    fopen.close()
    print("Data written in another file")

#writedata()
print("WRITING LOG IN INFORMATION IN A CSV FILE")
filename1=input("Enter a filename:")
writedata(filename1)

#displaydata()
print("READING AND DISPLAYING THE DATA FROM CSV FILE")
displaydata(filename1)

#writefile()
print("READING DATA AND WRITING IT INTO ANOTHER FILE")
filename2=input("Enter the other filename:")
writefile(filename1,filename2)

#reading filename2
print("Reading data from",filename2)
displaydata(filename2)

```

## Assignment 25

Create a module Number.py in python. The Number module has the following functions:

1. prime(m,n) which takes m and n as argument and print all the prime numbers between m and n

2. square(n) which takes n as argument and print the sum of all the numbers between 1 and n, which are divisible by 3 and 5.

3. Find\_num() which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 3000 (both included)

Write a program which imports Number.py module and uses the different functions defined in the module.

### OUTPUT:

```
Prime numbers between 40 and 50:  
41  
43  
47  
Sum of all numbers between 1 and 33 that are divisible by 3 and 5:  
45  
Find all numbers between 100 and 300 that are divisible by 7 but are not a multiple of 5:  
112, 119, 126, 133, 147, 154, 161, 168, 182, 189, 196, 203, 217, 224, 231, 238, 252, 259, 266, 273, 287, 294,
```

### CODE:

```
""" Number.py """  
# Functions  
def prime(m, n):  
    """ Prints all the prime numbers between m and n """  
    for i in range(m, n):  
        is_prime = True  
        for j in range(2, i):  
            if i % j == 0:  
                is_prime = False  
        if is_prime:  
            print(i)  
  
def square(n):  
    """ Prints the sum of all the numbers between 1 and n, which are divisible by 3 and 5 """  
    _sum = 0  
    for i in range(1, n):  
        if i % 15 == 0:  
            _sum += i  
    print(_sum)  
  
def find_num():  
    """ Finds all such numbers which are divisible by 7 but are not a multiple of 5, between  
    1000 and 3000 """  
    for i in range(100, 301):  
        if i % 7 == 0 and i % 5 != 0:  
            print(i, end=', ')
```

```
"""Runner.py"""
import Number

print("Prime numbers between 40 and 50:")
Number.prime(40, 50)

print("Sum of all numbers between 1 and 33 that are divisible by 3 and 5:")
Number.square(33)

print("Find all numbers between 100 and 300 that are divisible by 7 but are not a multiple
of 5:")
Number.find_num()
```

## Assignment 26

Create a module lengthconversion.py that stores functions for various length conversion:

1. Miletokm() - to convert miles to kilometer
2. Kmtomile() - to convert kilometer to miles
3. Feetinches - to convert feet to inches
4. Inchestofoot - to convert inches to feet

It should also store constant values such as value of(mile in kilometers and vice versa).

[1 mile=1.609344 kilometers; 1 feet = 12 inches]

Help() function should display proper information.

Import the above module in a program and show the output for different conversion based on user input.

### OUTPUT:

```
Help on module lengthconversion:

NAME
    lengthconversion - converts various lengths

FUNCTIONS
    Feetinches(value=1)
        # to convert feets to inches

    Inchestofeet(value=1)
        # to convert inches to feets

    Kmtomile(value=1)
        # to convert kilometers to miles

    Miletokm(value=1)
        # to convert miles to kilometers

DATA
    Feet = 12
    Inches = 0.08333333333333333
    Km = 0.621371
    Mile = 1.609344

FILE
    d:\program\record_file\lengthconversion.py

    Enter value to convert:3
    1:convert Kilometers to Miles
    2:convert Miles to Kilometers
    3:convert Feets to Inches
    4:convert Inches to Feets
    Enter choice:2
    3 Miles in Kilometers is 4.828032
    Do you want to convert more numbers? (y,n)n
```

### CODE:

```
'''lengthconversion.py'''
#lengthconversion
''' converts various lengths'''

#1 mile in kilometers
```

```
Mile = 1.609344
```

```
'''1 kilometer in miles'''
```

```
Km = 0.621371
```

```
Feet = 12#1 feet in inches
```

```
"""1 inche in feets"""
Inches = 1/12
```

```
# to convert miles to kilometers
```

```
def Miletokm(value=1):
    return value * Mile
```

```
# to convert kilometers to miles
```

```
def Kmtomile(value=1):
    return value * Km
```

```
# to convert feets to inches
```

```
def Feettoinches(value=1):
    return value * Feet
```

```
# to convert inches to feets
```

```
def Inchestofeet(value=1):
    return value * Inches
```

```
'''main.py'''
```

```
import lengthconversion
```

```
ch='y'
```

```
help(lengthconversion)
```

```
while ch=='y':
```

```
    num=int(input("Enter value to convert:"))
```

```
    print("1:convert Kilometers to Miles")
```

```
    print("2:convert Miles to Kilometers")
```

```
    print("3:convert Feets to Inches")
```

```
    print("4:convert Inches to Feets")
```

```
    choice=int(input("Enter choice:"))
```

```
    if(choice==1):
```

```
        print(num,"Kilometers in Miles is",lengthconversion.Kmtomile(num))
```

```
    elif(choice==2):
```

```
        print(num,"Miles in Kilometers is",lengthconversion.Miletokm(num))
```

```
    elif(choice==3):
```

```
        print(num,"Feets in Inches is",lengthconversion.Feettoinches(num))
```

```
    elif(choice==4):
```

```
        print(num,"Inches in Feets is",lengthconversion.Inchestofoot(num))
```

```
    ch=input("Do you want to convert more numbers? (y,n)")
```

## Assignment 27

Create a package Shape with the following modules:

1. Rectangle.py – includes functions for calculating area and perimeter
2. Square.py – includes functions for calculating area and perimeter
3. Triangle.py – includes functions for calculating area and perimeter
4. Circle.py – includes functions for calculating area and circumference

Help() function should display proper information.

Make sure that the above package meets the requirement of being a python package. Also, import above package and its modules using import command in two different programs with proper output statement.

### OUTPUT:

Area of all Shapes

```
>>>
= RESTART: C:/Users/Vinay Surana/Desktop/Python/Area of all Shapes in Module.py
Area of Circle is: 314.0
Area of Rectangle is: 20
Area of Triangle is: 10.0
Area of Square is: 25
>>> |
```

Perimeter of all Shapes

```
>>>
=====
RESTART: C:/Users/Vi
Perimeter of Circle is: 62.800000000000004
Perimeter of Rectangle is: 18
Perimeter of Square is: 20
Perimeter of Triangle is: 12
>>>
```

### CODE:

```
from shape import circle
from shape import rectangle
from shape import square
from shape import triangle
```

"""Printing Area of All Shapes"""

```
print("Area of Circle is:", circle.areaofcircle(10))
print("Area of Rectangle is:", rectangle.areaofrectangle(4,5))
print("Area of Triangle is:", triangle.areaoftriangle(4,5))
print("Area of Square is:", square.areaofsquare(5))
```

Perimeter:

```
from shape import circle
from shape import rectangle
from shape import square
from shape import triangle
```

""" Printing Perimeters of all Shapes"""

```
print("Perimeter of Circle is:", circle.periofcircle(10))
```

```
print("Perimeter of Rectangle is:", rectangle.periofrectangle(4,5))
print("Perimeter of Square is:", square.periofsquare(5))
print("Perimeter of Triangle is:", triangle.perioftriangle(3,4,5))
```

Circle:

```
""" Finding Area and Perimeter of Circle"""
def areaofcircle(r):
    return 3.14*r*r
```

```
def periofcircle(r):
    return 2*3.14*r
```

Rectangle:

```
"""Finding Area and Perimeter of a Rectangle"""
def areaofrectangle(l,b):
    return (l*b)
```

```
def periofrectangle(l,b):
    return (2*(l+b))
```

Square:

```
""" Finding Area and Perimeter of a Square"""
def areaofsquare(s):
    return s**2
```

```
def periofsquare(s):
    return 4*s
```

Triangle:

```
""" Finding Area and Perimeter of a Triangle"""
def areaoftriangle(h,ba):
    return (1/2)*h*ba
```

```
def perioftriangle(a,b,c):
    return a+b+c
```

**Assignment 28**

Write a program to implement a list of names of countries as stack.

**OUTPUT:**

```
1.add countries
2.display countries
enter choice (1/2)1
enter name of countries Afghanistan
do you want to continue?y
enter name of countries Albania
do you want to continue?y
enter name of countries Algeria
do you want to continue?y
enter name of countries U.S.A
do you want to continue?y
enter name of countries India
do you want to continue?y
enter name of countries Canada
do you want to continue?y
enter name of countries U.K
do you want to continue?y
enter name of countries Australia
do you want to continue?y
enter name of countries Combodia
do you want to continue?n
do you want to continue? (y/n)y
enter choice (1/2)2
the names of countries are :
Combodia
Australia
U.K
Canada
India
U.S.A
Algeria
Albania
Afghanistan
do you want to continue? (y/n)n
>>> |
```

**CODE :**

```
countries =[]
def Push_countries():
    ch='y'
    global stack
    while ch=='y' or ch=='Y':
        name=input('enter name of countries ')
        countries.append(name)
        ch=input('do you want to continue?')
        if ch=='n' or ch=='N':
            break

def display_countries():
    global stack
    st_len=len(countries)
    if st_len <=0 :
        print('empty stack!!!')
    else:
        print('the names of countries are :')
        i=len(countries)-1
        while (i>=0):
            print(countries[i])
            i =i-1
print('1.add countries')
print('2.display countries')
st='y'
while(st=='y' or st=='Y'):
    ch=int(input('enter choice (1/2)'))
    if ch==1:
        Push_countries()
    elif ch==2:
        display_countries()
    else:
        print('wrong choice')
st=input('do you want to continue? (y/n)')
```

## Assignment 29

Write a program to implement a nested list(username, password) as stack.

### OUTPUT:

```
Choice 1: Add Usernames and Passwords in a nested list to stack
Choice 2: Delete Username and Password from Stack
Choice 3: Display Stack

Enter your choice1

Enter usernamevivid
Enter passwordblues
Do you wish to add more usernames and passwords? Y/NY
Enter usernamesirusly
Enter passwordmalec
Do you wish to add more usernames and passwords? Y/NY
Enter usernameenigma
Enter passwordeejit
Do you wish to add more usernames and passwords? Y/NN

Do you want to continue? (Y/N)Y

Enter your choice2

Username and password deleted from the list are: ['enigma', 'eejit']

Do you want to continue? (Y/N)Y

Enter your choice3

The stack elements are:
['vivid', 'blues']
['sirusly', 'malec']

Do you want to continue? (Y/N)N

>>> |
```

### CODE:

```
stack=[]
#adding element into a stack
def push_value():
    ch='Y'
    global stack
    while ch=="y" or ch=="Y":
        a=input("Enter username")
        b=input("Enter password")
        stack.append([a,b]) #adding username and password
        ch=input("Do you wish to add more usernames and passwords? Y/N")
    if ch=="n" or ch=="N":
        break

def pop_value():
    global stack
```

```

x=len(stack) #returns the total number of elements in the stack
if x<=0: #checks for empty stack
    print("Underflow! Stack is empty")
else:
    element=stack.pop() #removing username and password from list
print("Username and password deleted from the list are:",element)

def display():
    global stack
    x=len(stack)
    if x<=0: #checks for empty stack
        print("Stack empty")
    else:
        print("The stack elements are:")
    for i in stack:
        print(i)

print("Choice 1: Add Usernames and Passwords in a nested list to stack")
print("Choice 2: Delete Username and Password from Stack")
print("Choice 3: Display Stack")
print()
c='y'
while c=='y' or c=='Y':
    ch=int(input("Enter your choice"))
    print()
    if ch==1:
        push_value()
    elif ch==2:
        pop_value()
    elif ch==3:
        display()
    else:
        print("Wrong Choice!! Please enter correct choice (1/2/3)")
    print()
    c=input("Do you want to continue? (Y/N)")
    print()

```

## Assignment – 30

Create the following table TEACHER and write SQL commands for (a) to (l):

No	Name	Age	Department	Date of Join	Salary	Sex
1.	Jigal	34	Computer	10/01/97	12000	M
2.	Sharmila	31	History	24/03/98	20000	F
3.	Sandeep	32	Maths	12/12/96	30000	M
4.	Sangeeta	35	History	01/07/99	40000	F
5.	Rakesh	42	Maths	05/09/97	25000	M
6.	Shyam	50	History	27/02/97	30000	M
7.	Shiv Om	44	Computer	25/02/97	21000	M
8.	Shalakha	33	Maths	31/07/97	20000	F

- a) To show all information about the teacher of History department.
- b) To list the names of female teachers who are in Maths department.
- c) To list names of all teachers with their date of joining in ascending order.
- d) To display name, salary, age for male teacher only.
- e) To count the number of teachers with age>23.
- f) To insert a new row in the TEACHER table with the following data:  
9,"SANA", 26, "COMPUTER", 13/05/95,23000,'F'
- g) Add a new column address
- h) Arrange the whole table in alphabetical order to name.
- i) Display the department and salary of teachers whose name starts with 'S'.
- j) To increase the salary of all the teachers of History department by 2000.
- k) Display the maximum and average salary department wise.
- l) Display the details of all the teachers who were hired during 1997.

Original Table

```
mysql> SELECT * FROM TEACHER;
+---+-----+-----+-----+-----+-----+-----+
| No | Name  | Age   | Department | DateofJoin | Salary | Sex  |
+---+-----+-----+-----+-----+-----+-----+
| 1  | Jigal | 34    | Computer   | 1997-01-10 | 12000 | M   |
| 2  | Sharmila | 31    | History    | 1998-03-24 | 20000 | F   |
| 3  | Sandeep | 32    | Maths      | 1996-12-12 | 30000 | M   |
| 4  | Sangeeta | 35    | History    | 1999-07-01 | 40000 | F   |
| 5  | Rakesh | 42    | Maths      | 1997-09-05 | 25000 | M   |
| 6  | Shyam | 50    | History    | 1997-02-27 | 30000 | M   |
| 7  | Shiv Om | 44    | Computer   | 1997-02-25 | 21000 | M   |
| 8  | Shalakha | 33    | Maths      | 1997-07-31 | 20000 | F   |
+---+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

- a) To show all information about the teacher of History department

```
mysql> SELECT * FROM TEACHER WHERE Department='History';
+---+-----+-----+-----+-----+-----+-----+
| No | Name  | Age   | Department | DateofJoin | Salary | Sex  |
+---+-----+-----+-----+-----+-----+-----+
| 2  | Sharmila | 31    | History    | 1998-03-24 | 20000 | F   |
| 4  | Sangeeta | 35    | History    | 1999-07-01 | 40000 | F   |
| 6  | Shyam | 50    | History    | 1997-02-27 | 30000 | M   |
+---+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

b) To list the names of female teachers who are in Maths department

```
mysql> SELECT * FROM TEACHER WHERE Sex='F' AND Department='Maths';
+---+-----+-----+-----+-----+-----+
| No | Name   | Age   | Department | DateofJoin | Salary | Sex |
+---+-----+-----+-----+-----+-----+
| 8  | Shalakha | 33   | Maths      | 1997-07-31 | 20000 | F   |
+---+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

c) To list names of all teachers with their date of joining in ascending order

```
mysql> SELECT Name,DateofJoin FROM TEACHER ORDER BY DateofJoin;
+-----+-----+
| Name    | DateofJoin |
+-----+-----+
| Sandeep | 1996-12-12 |
| Jigal   | 1997-01-10 |
| Shiv Om | 1997-02-25 |
| Shyam   | 1997-02-27 |
| Shalakha | 1997-07-31 |
| Rakesh  | 1997-09-05 |
| Sharmila | 1998-03-24 |
| Sangeeta | 1999-07-01 |
+-----+-----+
8 rows in set (0.02 sec)

mysql>
```

d) To display name, salary, age for male teacher only

```
mysql> SELECT Name,Salary,Age FROM TEACHER WHERE Sex='M';
+-----+-----+-----+
| Name   | Salary | Age  |
+-----+-----+-----+
| Jigal  | 12000 | 34   |
| Sandeep | 30000 | 32   |
| Rakesh | 25000 | 42   |
| Shyam  | 30000 | 50   |
| Shiv Om | 21000 | 44   |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

e) To count the number of teachers with age>23

```
mysql> SELECT COUNT(*) FROM TEACHER WHERE Age>23;
+-----+
| COUNT(*) |
+-----+
|     8    |
+-----+
1 row in set (0.02 sec)

mysql>
```

f) To insert a new row in the TEACHER table with the following data:

```
9,"SANA", 26, "COMPUTER", 13/05/95,23000,'F'  
mysql> INSERT INTO TEACHER VALUES(9, 'SANA', 26, 'COMPUTER', '1995/05/13', 23000, 'F');  
Query OK, 1 row affected (0.02 sec)  
  
mysql> SELECT * FROM TEACHER;  
+---+---+---+---+---+---+---+  
| No | Name | Age | Department | DateofJoin | Salary | Sex |  
+---+---+---+---+---+---+---+  
| 1 | Jigal | 34 | Computer | 1997-01-10 | 12000 | M |  
| 2 | Sharmila | 31 | History | 1998-03-24 | 20000 | F |  
| 3 | Sandeep | 32 | Maths | 1996-12-12 | 30000 | M |  
| 4 | Sangeeta | 35 | History | 1999-07-01 | 40000 | F |  
| 5 | Rakesh | 42 | Maths | 1997-09-05 | 25000 | M |  
| 6 | Shyam | 50 | History | 1997-02-27 | 30000 | M |  
| 7 | Shiv Om | 44 | Computer | 1997-02-25 | 21000 | M |  
| 8 | Shalakha | 33 | Maths | 1997-07-31 | 20000 | F |  
| 9 | SANA | 26 | COMPUTER | 1995-05-13 | 23000 | F |  
+---+---+---+---+---+---+---+  
9 rows in set (0.00 sec)  
  
mysql>
```

g) Add a new column address

```
mysql> ALTER TABLE TEACHER ADD ADDRESS VARCHAR(40);  
Query OK, 0 rows affected (0.55 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
mysql> SELECT * FROM TEACHER;  
+---+---+---+---+---+---+---+---+  
| No | Name | Age | Department | DateofJoin | Salary | Sex | ADDRESS |  
+---+---+---+---+---+---+---+---+  
| 1 | Jigal | 34 | Computer | 1997-01-10 | 12000 | M | Jaipur |  
| 2 | Sharmila | 31 | History | 1998-03-24 | 20000 | F | Dispur |  
| 3 | Sandeep | 32 | Maths | 1996-12-12 | 30000 | M | Pune |  
| 4 | Sangeeta | 35 | History | 1999-07-01 | 40000 | F | Ahmedabad |  
| 5 | Rakesh | 42 | Maths | 1997-09-05 | 25000 | M | Delhi |  
| 6 | Shyam | 50 | History | 1997-02-27 | 30000 | M | Agra |  
| 7 | Shiv Om | 44 | Computer | 1997-02-25 | 21000 | M | Mumbai |  
| 8 | Shalakha | 33 | Maths | 1997-07-31 | 20000 | F | Surat |  
| 9 | SANA | 26 | COMPUTER | 1995-05-13 | 23000 | F | Gandhinagar |  
+---+---+---+---+---+---+---+---+  
9 rows in set (0.00 sec)  
  
mysql>
```

h) Arrange the whole table in alphabetical order to name

```
mysql> SELECT * FROM TEACHER ORDER BY Name;  
+---+---+---+---+---+---+---+---+  
| No | Name | Age | Department | DateofJoin | Salary | Sex | ADDRESS |  
+---+---+---+---+---+---+---+---+  
| 1 | Jigal | 34 | Computer | 1997-01-10 | 12000 | M | Jaipur |  
| 5 | Rakesh | 42 | Maths | 1997-09-05 | 25000 | M | Delhi |  
| 9 | SANA | 26 | COMPUTER | 1995-05-13 | 23000 | F | Gandhinagar |  
| 3 | Sandeep | 32 | Maths | 1996-12-12 | 30000 | M | Pune |  
| 4 | Sangeeta | 35 | History | 1999-07-01 | 40000 | F | Ahmedabad |  
| 8 | Shalakha | 33 | Maths | 1997-07-31 | 20000 | F | Surat |  
| 2 | Sharmila | 31 | History | 1998-03-24 | 20000 | F | Dispur |  
| 7 | Shiv Om | 44 | Computer | 1997-02-25 | 21000 | M | Mumbai |  
| 6 | Shyam | 50 | History | 1997-02-27 | 30000 | M | Agra |  
+---+---+---+---+---+---+---+---+  
9 rows in set (0.00 sec)  
  
mysql>
```

i) Display the department and salary of teachers whose name starts with 'S'.

```
mysql> SELECT Department,Salary FROM TEACHER WHERE Name LIKE 'S%';  
+---+---+  
| Department | Salary |  
+---+---+  
| History | 20000 |  
| Maths | 30000 |  
| History | 40000 |  
| History | 30000 |  
| Computer | 21000 |  
| Maths | 20000 |  
| COMPUTER | 23000 |  
+---+---+  
7 rows in set (0.00 sec)  
  
mysql>
```

j) To increase the salary of all the teachers of History department by 2000

```
mysql> UPDATE TEACHER SET Salary=Salary+2000 WHERE Department='History';
Query OK, 3 rows affected (0.02 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

```
mysql> SELECT * FROM TEACHER;
```

No	Name	Age	Department	DateofJoin	Salary	Sex	ADDRESS
1	Jigal	34	Computer	1997-01-10	12000	M	Jaipur
2	Sharmila	31	History	1998-03-24	22000	F	Dispur
3	Sandeep	32	Maths	1996-12-12	30000	M	Pune
4	Sangeeta	35	History	1999-07-01	42000	F	Ahmedabad
5	Rakesh	42	Maths	1997-09-05	25000	M	Delhi
6	Shyam	50	History	1997-02-27	32000	M	Agra
7	Shiv Om	44	Computer	1997-02-25	21000	M	Mumbai
8	Shalakha	33	Maths	1997-07-31	20000	F	Surat
9	SANA	26	COMPUTER	1995-05-13	23000	F	Gandhinagar

```
9 rows in set (0.00 sec)
```

```
mysql>
```

k) Display the maximum and average salary department wise

```
mysql> SELECT Department,MAX(Salary),AVG(Salary) FROM TEACHER GROUP BY Department;
+-----+-----+-----+
| Department | MAX(Salary) | AVG(Salary) |
+-----+-----+-----+
| Computer   |      23000 |    18666.6667 |
| History    |      42000 |    32000.0000 |
| Maths       |      30000 |    25000.0000 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql>
```

l) Display the details of all the teachers who were hired during 1997

```
mysql> SELECT * FROM TEACHER WHERE DateofJoin BETWEEN '1997/01/01' AND '1997/12/31';
+-----+-----+-----+-----+-----+-----+-----+
| No   | Name  | Age   | Department | DateofJoin | Salary | Sex   | ADDRESS |
+-----+-----+-----+-----+-----+-----+-----+
| 1    | Jigal | 34    | Computer   | 1997-01-10 | 12000 | M    | Jaipur |
| 5    | Rakesh | 42    | Maths      | 1997-09-05 | 25000 | M    | Delhi  |
| 6    | Shyam  | 50    | History    | 1997-02-27 | 32000 | M    | Agra   |
| 7    | Shiv Om | 44    | Computer   | 1997-02-25 | 21000 | M    | Mumbai |
| 8    | Shalakha | 33    | Maths      | 1997-07-31 | 20000 | F    | Surat  |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql>
```

## Assignment – 31

Create the following relations EMPLOYEES and EMPSALARY. On the basis of these two tables write SQL commands for (a) to (d)

**EMPLOYEES**

EMPI D	FIRSTNA ME	LASTNAME	ADDRESS	CITY
010	George	Smith	83 First Street	Howard
105	Mary	Jones	842 Vine Ave	Losantiville
152	Sam	Tones	33 ELM St.	Paris
215	Sarah	Ackerman	440 U.S. 110	Upton
244	Manila	Sengupta	24 Friends Street	New Delhi
300	Robert	Samuel	9 Fifth Cross	Washington
335	Henry	Williams	12 Moore Street	Boston
400	Rachel	Lee	121 Harrison St.	New York
441	Peter	Thompson	11 Red Road	Paris

**EMPSALARY**

EMPID	SALARY	BENEFITS	DESIGNATION
010	75000	15000	Manager
105	65000	15000	Manager
152	80000	25000	Director
215	75000	12500	Manager
244	50000	12000	Clerk
300	45000	10000	Clerk
335	40000	10000	Clerk
400	32000	7500	Salesman
441	28000	7500	Salesman

```
create table EMPLOYEES(EMPID int primary key, FIRSTNAME varchar(50), LASTNAME varchar(50), ADDRESS varchar(100), CITY varchar(50));

insert into EMPLOYEES values(010,'George', 'Smith', '83 First street', 'Howard');
insert into EMPLOYEES values(105,'Mary', 'Jones', '842 Vine Ave', 'Losantiville');
insert into EMPLOYEES values(152,'Sam', 'Tones', '33 ELM St.', 'Paris');
insert into EMPLOYEES values(215,'Sarah', 'Ackerman', '440 U.S. 110', 'Upton');
insert into EMPLOYEES values(244,'Manila', 'Sengupta', '24 Friends Street', 'New Delhi');
insert into EMPLOYEES values(300,'Robert', 'Samuel', '9 Fifth Cross', 'Washington');
insert into EMPLOYEES values(335,'Henry', 'Williams', '12 Moore Street', 'Boston');
insert into EMPLOYEES values(400,'Rachel', 'Lee', '121 Harrison St.', 'New York');
insert into EMPLOYEES values(441,'Peter', 'Thompson', '11 Red Road', 'Paris');
```

010	George	Smith	83 First street	Howard
105	Mary	Jones	842 Vine Ave	Losantiville
152	Sam	Tones	33 ELM St.	Paris
215	Sarah	Ackerman	440 U.S. 110	Upton
244	Manila	Sengupta	24 Friends Street	New Delhi
300	Robert	Samuel	9 Fifth Cross	Washington
335	Henry	Williams	12 Moore Street	Boston
400	Rachel	Lee	121 Harrison St.	New York
441	Peter	Thompson	11 Red Road	Paris

```

create table EMPSALARY(EMPID int primary key, SALARY int, BENEFITS int, DESIGNATION varchar(50));

insert into EMPSALARY values(010,75000,15000,'Manager');
insert into EMPSALARY values(105,65000,15000,'Manager');
insert into EMPSALARY values(152,80000,25000,'Director');
insert into EMPSALARY values(215,75000,12500,'Manager');
insert into EMPSALARY values(244,50000,12000,'Clerk');
insert into EMPSALARY values(300,45000,10000,'Clerk');
insert into EMPSALARY values(335,40000,10000,'Clerk');
insert into EMPSALARY values(400,32000,7500,'Salesman');
insert into EMPSALARY values(441,28000,7500,'Salesman');

```

010	75000	15000	Manager
105	65000	15000	Manager
152	80000	25000	Director
215	75000	12500	Manager
244	50000	12000	Clerk
300	45000	10000	Clerk
335	40000	10000	Clerk
400	32000	7500	Salesman
441	28000	7500	Salesman

- a) To display the details of those employees living in Paris from tables EMPLOYEES and EMPSALARY.

```
select * from EMPLOYEES,EMPSALARY where CITY='Paris' and EMPLOYEES.EMPID=EMPSALARY.EMPID;
```

152	Sam	Tones	33 ELM St.	Paris	152	80000	25000	Director
441	Peter	Thompson	11 Red Road	Paris	441	28000	7500	Salesman

- b) To display the Average Salary designation wise from the table EMPSALARY.

```
select avg(SALARY),DESIGNATION from EMPSALARY group by DESIGNATION;
```

45000.0	Clerk
80000.0	Director
71666.6666666667	Manager
30000.0	Salesman

c) To display the minimum salary among managers from the table EMPSALARY.

```
select min(SALARY) from EMPSALARY where DESIGNATION='Manager';
```

-----
65000
-----

d) To display the employee details from the table EMPLOYEES who are from New Delhi.

```
select * from EMPLOYEES where CITY='New Delhi';
```

-----	-----	-----	-----	-----
244	Manila	Sengupta	24 Friends Street	New Delhi
-----	-----	-----	-----	-----

e) Count the number of employee having designation as 'Manager'

```
select count(DESIGNATION) from EMPSALARY where DESIGNATION='Manager';
```

---
3
---

f) To display First name, Last name and total salary of all clerks where total salary is calculated as SALARY + BENEFITS

```
select FIRSTNAME, LASTNAME, SALARY+BENEFITS from EMPLOYEES, EMPSALARY where DESIGNATION='Clerk' and EMPLOYEES.EMPID=EMPSALARY.EMPID;
```

-----	-----	-----
Manila	Sengupta	62000
-----	-----	-----
Robert	Samuel	55000
-----	-----	-----
Henry	Williams	50000
-----	-----	-----

g) Display EMPID and ADDRESS of all Clerk whose SALARY is equal to 10000.

```
select EMPSALARY.EMPID, ADDRESS from EMPLOYEES, EMPSALARY where DESIGNATION='Clerk' and SALARY=10000 and EMPLOYEES.EMPID=EMPSALARY.EMPID;
```

### Assignment 32

Write a program to connect to employee table in SQL and do the following on user choice:

- i. Fetch all records and display
- ii. Delete a record from the table
- iii. Insert a new record to the table
- iv. Update a record value

#### OUTPUT:

```
What Would You Like To Do?  
1-Fetch all records and display them  
2>Delete a record from the table  
3-Insert a new record to the table  
4-Update a record value  
  
enter your choice: 1  
  
(101, 'Sharma', 'Amit', 'Sales')  
(102, 'Arora', 'Shiv', 'Personnel')  
(103, 'Lakshmi', 'KS', 'Accounts')  
(104, 'Sales', 'Shivika', 'Accounts')  
(105, 'Tharkal', 'Savtik', 'Sales')
```

```
In [2]: runfile('C:/Users/ICTLAB5-9/Documents/sqlconnectivity.  
What Would You Like To Do?  
1-Fetch all records and display them  
2>Delete a record from the table  
3-Insert a new record to the table  
4-Update a record value  
  
enter your choice: 2  
  
enter the ecode of the record you would like to delete: 103  
(101, 'Sharma', 'Amit', 'Sales')  
(102, 'Arora', 'Shiv', 'Personnel')  
(104, 'Sales', 'Shivika', 'Accounts')  
(105, 'Tharkal', 'Savtik', 'Sales')
```

```
In [3]: runfile('C:/Users/ICTLAB5-9/Documents/sqlconnectivity.  
What Would You Like To Do?  
1-Fetch all records and display them  
2>Delete a record from the table  
3-Insert a new record to the table  
4-Update a record value  
  
enter your choice: 3  
  
Enter the following details  
  
Enter the ecode: 103
```

```

Enter lastname: Lakshmi

Enter firstname: KS

Enter department: Accounts
(101, 'Sharma', 'Amit', 'Sales')
(102, 'Arora', 'Shiv', 'Personnel')
(103, 'Lakshmi', 'KS', 'Accounts')
(104, 'Sales', 'Shivika', 'Accounts')
(105, 'Tharkal', 'Savtik', 'Sales')

```

**CODE:**

```

import mysql.connector as sqltor
print('What Would You Like To Do?')
print('1-Fetch all records and display them')
print('2>Delete a record from the table')
print('3-Insert a new record to the table')
print('4-Update a record value')
op=int(input('enter your choice: '))
print()
if(op==1):
    mycon=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
    cursor=mycon.cursor()
    cursor.execute("select * from employee")
    data=cursor.fetchall()
    for i in data:
        print(i)
    mycon.close()
elif(op==2):
    x=input('enter the ecode of the record you would like to delete: ')
    my=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
    cur=my.cursor()
    cur.execute("DELETE FROM employee WHERE ecode=%s",[x])
    cur.execute("select * from employee")
    data=cur.fetchall()
    for i in data:
        print(i)
    my.commit()
    my.close()
elif(op==3):
    print('Enter the following details')
    ec=int(input('Enter the ecode: '))
    ln=input('Enter lastname: ')

```

```

fn=input('Enter firstname: ')
d=input('Enter department: ')
m=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
c=m.cursor()
query=("insert into employee
values(%(Ecode)s,%(Lastname)s,%(Firstname)s,%(Department)s)")
data={'Ecode':ec,'Lastname':ln,'Firstname':fn,'Department':d}
c.execute(query,data)
c.execute("select * from employee")
data=c.fetchall()
for i in data:
    print(i)
m.commit()
m.close()

elif(op==4):
    ec=int(input('Enter the ecode of the entry you want to update'))
    print('1-Lastname')
    print('2-Firstname')
    print('3-Department')
    choice=int(input('Enter the option number of the column you want to update'))
    if(choice==1):
        myc=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
        cu=myc.cursor()
        ln=input('Enter value to be updated: ')
        cu.execute('UPDATE employee SET Lastname=%s WHERE ecode=%s',(ln,ec))
        myc.commit()
        myc.close()
    elif(choice==2):
        fn=input('Enter value to be updated: ')
        myconn=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
        curs=myconn.cursor()
        curs.execute('UPDATE employee SET Firsttname=%s WHERE ecode=%s',(fn,ec))
        myconn.commit()
        myconn.close()
    elif(choice==3):
        d=input('Enter value to be updated: ')
        myconne=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
        curso=myconne.cursor()
        curso.execute('UPDATE employee SET Department=%s WHERE ecode=%s',(d,ec))
        myconne.commit()

```

```
myconne.close()
n=sqltor.connect(host="localhost",user="root",password="",database="emp_data")
a=n.cursor()
a.execute("select * from employee")
data=a.fetchall()
for i in data:
    print(i)
n.close()

else:
    print('Please enter valid option')
```